Jurnal Pendidikan dan Teknologi Indonesia (JPTI)
Vol. 5, No. 12, Desember 2025, Hal. 3610-3619
DOI: https://doi.org/10.52436/1.jpti.1197
p-ISSN: 2775-4227
e-ISSN: 2775-4219

Deteksi dan Klasifikasi Ancaman pada Log Serangan Siber Menggunakan Algoritma K-Nearest Neighbor (KNN) dan Random Forest (RF)

Aissyah Wahyu Ningrum[*1], Mukhlis Prasetyo Aji[2], Ermadi Satriya Wijaya[3], Elindra Ambar Pambudi[4]

[1,2,3,4]Teknik Informatika, Fakultas Teknik dan Sains, Universitas Muhammadiyah Purwokerto, Indonesia
Email: [1]aissyahwn@gmail.com, [2]prasetyo-aji@ump.ac.id, [3]ermadi_satriya@ump.ac.id, [4]elindraambarpambudi@ump.ac.id

Abstrak

Ancaman siber yang semakin kompleks dan terus berkembang menuntut sistem keamanan yang mampu mendeteksi serangan secara cepat dan akurat. Pesatnya perkembangan serangan siber menuntut sistem deteksi yang cerdas dan adaptif untuk mengamankan jaringan informasi. Penelitian ini bertujuan untuk menerapkan dan mengevaluasi kinerja algoritma *K-Nearest Neighbor* (KNN) dan *Random Forest* (RF) dalam mendeteksi serta mengklasifikasikan ancaman berdasarkan log serangan siber. Data yang digunakan diperoleh dari Biro Sistem Informasi Universitas Muhammadiyah Purwokerto, berjumlah 500 entri dengan 25 atribut, yang kemudian diproses melalui tahap pra-pemrosesan seperti parsing, imputasi nilai hilang, dan *encoding* atribut kategorikal. Model KNN dan RF dibangun dan diuji menggunakan metrik evaluasi akurasi, *precision*, *recall*, dan *f1-score*. Hasil menunjukkan bahwa algoritma RF memiliki kinerja yang lebih unggul dengan akurasi 94,87% dibandingkan KNN yang mencapai 89,32%. Selain itu, RF menunjukkan konsistensi tinggi dalam *precision* dan *recall* pada kedua kelas, menjadikannya lebih efektif dalam mendeteksi variasi serangan. Dengan demikian, RF direkomendasikan sebagai algoritma utama dalam pengembangan sistem deteksi ancaman siber berbasis pembelajaran mesin.

Kata kunci: *Deteksi, Keamanan, K-Nearest Neighbor (KNN), Random Forest (RF), Siber*

Threat Detection and Classification in Cyber Attack Logs Using K-Nearest Neighbor (KNN) and Random Forest (RF) Algorithms

Abstract

The increasingly complex and evolving cyber threats demand a security system that is able to detect attacks quickly and accurately. The rapid development of cyber attacks demands an intelligent and adaptive detection system to secure information networks. This study aims to implement and evaluate the performance of the K-Nearest Neighbor (KNN) and Random Forest (RF) algorithms in detecting and classifying threats based on cyber attack logs. The data used were obtained from the Information Systems Bureau of Muhammadiyah University of Purwokerto, totaling 500 entries with 25 attributes, which were then processed through pre-processing stages such as parsing, missing value imputation, and categorical attribute encoding. The KNN and RF models were built and tested using evaluation metrics of accuracy, precision, recall, and f1-score. The results show that the RF algorithm has superior performance with an accuracy of 94.87% compared to KNN which reached 89.32%. In addition, RF shows high consistency in precision and recall across both classes, making it more effective in detecting variations in attacks. Thus, RF is recommended as the main algorithm in the development of a machine learning-based cyber threat detection system.

Keywords: *Cyber, Detection, K-Nearest Neighbor (KNN), Random Forest (RF), Security*

1. PENDAHULUAN

Kemajuan Artificial Intelligence (AI) telah mendorong peningkatan efektivitas pendekatan berbasis pembelajaran dalam mendeteksi serangan siber [1]. Namun, dinamika dan kompleksitas serangan siber yang terus berkembang menyebabkan perlindungan terhadap sistem teknologi informasi (TI) dari berbagai ancaman dan aktivitas berbahaya dalam jaringan masih menjadi tantangan besar [2]. Ancaman keamanan melalui jaringan

p-ISSN: 2775-4227
e-ISSN: 2775-4219

memerlukan kewaspadaan konstan terhadap perangkat penyerang yang terus berkembang, karena penyerang secara aktif meningkatkan teknik untuk mengeksploitasi celah keamanan yang kerap muncul sejak tahap awal pengembangan perangkat lunak, seperti peretasan, pencurian data, hingga serangan *denial-of-service* (DoS) [3]. Serangan tersebut umumnya meninggalkan jejak digital dalam log sistem, yang apabila dianalisis secara tepat dapat memberikan informasi penting mengenai pola serangan dan potensi ancaman keamanan [4].

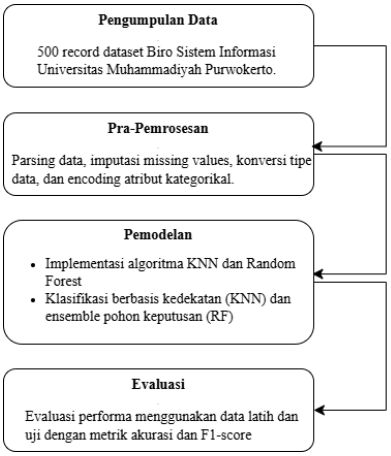
Log serangan siber merupakan sumber data yang penting dalam mendeteksi dan mengklasifikasikan ancaman karena memuat informasi aktivitas jaringan secara rinci [5]. Namun, tingginya volume dan kompleksitas data log sering kali menyulitkan proses analisis manual, sehingga menimbulkan kebutuhan akan pendekatan otomatis. Untuk itu, pemanfaatan kecerdasan buatan dan algoritma pembelajaran mesin menjadi solusi yang efektif dalam mengolah dan mengekstraksi informasi dari data log secara efisien [6]. Pemanfaatan pembelajaran mesin sebagai bagian dari teknologi analisis ancaman cerdas digunakan untuk meningkatkan akurasi dan kecepatan dalam deteksi dan klasifikasi ancaman siber [7]. Proses deteksi bertujuan untuk mengenali keberadaan aktivitas anomali dalam jaringan, seperti pola lalu lintas yang tidak biasa atau upaya akses tidak sah [8]. Sementara itu, klasifikasi berfungsi untuk mengelompokkan ancaman berdasarkan karakteristik tertentu, seperti sumber serangan, jenis teknik yang digunakan, atau target yang disasar [9].

Deteksi serangan telah dilakukan oleh beberapa penelitian sebelumnya, salah satunya adalah penelitian oleh Maulana dan Alamsyah (2023) yang berjudul "Optimalisasi Deteksi Serangan DDoS Menggunakan Algoritma Random Forest, SVM, KNN dan MLP pada Jaringan Komputer." Penelitian ini menunjukkan bahwa algoritma *Random Forest* memiliki akurasi prediksi tertinggi sebesar 99,41%, diikuti oleh K-Nearest Neighbor (99%), SVM (98,37%), dan MLP (93,97%). Temuan tersebut menunjukkan bahwa algoritma *Random Forest* dan KNN unggul dalam proses klasifikasi serangan dibandingkan metode lainnya. Oleh karena itu, penelitian ini menggunakan dua metode tersebut, yaitu *Random Forest* dan KNN, sebagai dasar dalam deteksi serangan [10]. Pemilihan *Random Forest* (RF) didasarkan pada kestabilan performa, interpretabilitas, dan efisiensi komputasi pada data berskala menengah. Metode deep learning belum digunakan karena keterbatasan data, kebutuhan akan hasil yang cepat, serta minimnya infrastruktur komputasi, sehingga RF dinilai lebih sesuai untuk konteks penelitian ini.

Metode K-Nearest Neighbor (KNN) memiliki beberapa kelebihan yaitu tangguh terhadap training data yang noisy dan efektif apabila training data-nya besar [11]. Sementara *Random Forest* (RF) memiliki kelebihan dalam proses iterasi komputasi yang lebih cepat [12]. Berdasarkan hal tersebut, tujuan penelitian ini adalah untuk menerapkan dan mengevaluasi kinerja kedua algoritma KNN dan RF dalam mendeteksi dan mengklasifikasikan ancaman pada log serangan siber secara lebih optimal. Penelitian ini akan menganalisis sejauh mana algoritma KNN dan RF mampu mengidentifikasi pola serangan secara akurat dan efisien pada dataset log jaringan yang kompleks. Diharapkan dengan adanya penelitian ini, dapat diperoleh model deteksi ancaman yang lebih andal, yang tidak hanya meningkatkan akurasi dalam identifikasi serangan, tetapi juga dapat digunakan sebagai referensi pengembangan sistem keamanan jaringan berbasis pembelajaran mesin di masa depan.

2. METODE PENELITIAN

Penelitian ini dilakukan melalui tahapan-tahapan yang tersusun secara sistematis, dimulai dari proses pengumpulan data, dilanjutkan dengan tahap praproses data, pemodelan, hingga evaluasi kinerja model guna memastikan hasil yang diperoleh memiliki tingkat akurasi dan relevansi yang tinggi, seperti yang ditunjukkan pada Gambar 1 [13].



Gambar 1. Tahapan Penelitian

p-ISSN: 2775-4227
e-ISSN: 2775-4219

2.1 Pengumpulan Data

Penelitian ini menggunakan dataset yang diperoleh dari Biro Sistem Informasi Universitas Muhammadiyah Purwokerto, yang menyediakan data khusus untuk mendeteksi intrusi siber. Seluruh data dilabeli untuk mengindikasikan jenis ancaman, sehingga dapat langsung digunakan dalam proses pelatihan dan pengujian model klasifikasi menggunakan algoritma *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF).

2.2 Pra-Pemrosesan

Tahap pre-processing merupakan proses penyiapan data mentah agar siap digunakan dalam analisis dan pemodelan [14]. Dataset yang diperoleh dari Biro Sistem Informasi Universitas Muhammadiyah Purwokerto terdiri dari 500 entri dalam satu kolom dengan atribut-atribut yang dipisahkan oleh tanda titik koma (;). pemisahan data ke dalam atribut yang sesuai menggunakan *delimiter-based parsing*. Setelah data dipisahkan, dilakukan pemgecekan *missing values* dan inputasi jika ditemukan data yang hilang. Selanjutnya, dilakukan konversi tipe data, di mana atribut numerik yang terbaca sebagai teks dikonversi ke format yang sesuai. Atribut kategorikal seperti *Protocol, Method, dan Action* juga dikodekan menggunakan *Lebel Encoding atau One-Hot Encoding* agar dapat diolah oleh model pembelajaran mesih. Rangkaian dataset di semua proses ini menjadi lebih terstruktur dan layak untuk digunakan dalam pelatihan model deteksi ancaman siber [10].

2.3 Pemodelan

Penelitian ini menggunakan dua algoritma, yaitu *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF):

1. *K-Nearest Neighbors* (KNN)

Algoritma *K-Nearest Neighbors* (KNN) adalah salah satu metode dalam machine learning yang menerapkan pendekatan pembelajaran terawasi *(supervised learning*). KNN populer karena konsepnya yang sederhana dan mudah diterapkan dalam berbagai permasalahan klasifikasi maupun regresi. Algoritma ini tergolong ke dalam tipe pembelajar malas (*lazy learner*), karena tidak membentuk model secara eksplisit selama proses pelatihan, melainkan langsung melakukan prediksi berdasarkan data latih yang tersedia saat proses pengujian. Oleh karena itu, KNN dianggap sebagai metode klasifikasi yang sederhana dan mudah dipahami [15]. Dalam proses klasifikasinya, objek baru diklasifikasikan berdasarkan kedekatan jaraknya dengan data pelatihan terdekat [16]. KNN bekerja dengan cara memantau (monitoring) hasil *query* yang kemudian diklasifikasikan berdasarkan mayoritas kategori dari data tetangganya yang terdekat [17]. Tujuan utama metode ini adalah untuk mengklasifikasikan objek baru dengan mempertimbangkan atribut yang dimiliki serta sampel pelatihan yang tersedia [18]. Untuk menentukan kedekataan antar data, *K-Nearest Neighbors (KNN)* sering menggunakan metrik jarak *Euclidean*, ditunjukkan pada Rumus 1 [19].

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_{i-}\, p_i)} \qquad (1)$$

Keterangan :

- *d (p,q)* : jarak antara dua titik data *p* dan *q*.
- *n* : jumlah fitur dalam dataset.
- p_i & q_i : nilai atribut ke–*i* dari masing-masing titik data.

Proses Klasifikasi dengan KNN:

1. Menentukan jumlah tetangga terdekat *k.*
2. Menghitung jarak antara data uji dan semua data latih menggunakan rumus jarak (Euclidean).
3. Mengurutkan hasil perhitungan jarak dan memilih k tetangga dengan jarak terdekat.
4. Menentukan kelas mayoritas dari k tetangga terdekat tersebut.
5. Mengklasifikasikan data uji ke dalam kelas mayoritas tersebut.

2. *Random Forest* (RF)

Random Forest (RF) merupakan salah satu metode klasifikasi yang efektif untuk menangani data dalam jumlah besar [20]. Proses klasifikasi dalam algoritma ini dilakukan melalui penggabungan sejumlah pohon keputusan (*decision tree*) yang masing-masing dilatih menggunakan sampel data secara acak. Struktur pohon dalam RF terdiri atas beberapa elemen penting, yakni simpul akar (*root node*), simpul internal (*internal node*), serta simpul daun (*leaf node*) yang merupakan titik akhir dari pohon dengan satu jalur masuk tanpa jalur keluar [21]. Salah satu tahap penting dalam pembangunan pohon keputusan adalah penghitungan *entropy*, yang digunakan untuk mengukur tingkat ketidakteraturan atau ketidakmurnian suatu atribut dalam dataset. Nilai *entropy* ini menjadi dasar dalam menentukan information gain atau perolehan Proses pembentukan pohon dimulai dengan

p-ISSN: 2775-4227
e-ISSN: 2775-4219

menghitung nilai *entropy* sebagaimana dirumuskan oleh Rahayu et al. (2020) [22]. Berikut adalah beberapa rumus dan konsep kunci dalam algoritma Rendom Forest [23] :

1. Bootstrap sampling
   Random Forest dimulaih dengan mengambil sampel acak dari dataset asli menggunakan teknik bootstrap. Setiap subset yang diambil digunakan untuk melatih satu pohon keputusan.
2. Pembentukan *Decision trees*
   Setiap subset digunakan untuk membangun decision tree tanpa pemangkasan (pruning). Dalam proses ini, dataset log serangan siber yang terdiri dari 500 entri diacak dan dibagi menjadi beberapa subset kecil. Misalnya, lima sampel acak diambil dari data yang berisi atribut seperti jenis serangan, alamat IP sumber, waktu serangan, dan protokol yang digunakan.
3. *Voting* (Klasifikasi) atau *Averaging* (Regresi):
   1) Untuk klasifikasi, prediksi akhir didasarkan pada suara mayoritas dari semua *decision tree*.
   2) Untuk regresi, hasil akhir dihitung sebagai rata-rata dari semua prediksi *decision tree*.
4. Rumus Prediksi
   Untuk klasifikasi, rumus yang digunakan dapat dilihat pada Rumus 2.

$$\widehat{y} = mode\ (y_1, y_2, \dots, y_m) \qquad (2)$$

Keterangan:
$y_1, y_2, \dots, y_m$: hasil prediksi dari m *decision tree* dalam *Random Forest*.

2.4 Evaluasi

Evaluasi dilakukan dengan membandingkan hasil prediksi dari ketiga algoritma memanfaatkan sejumlah metrik seperti *precision, recall*, *F1-score*, dan *accuracy*. *Accuracy* mengukur tingkat kesesuaian prediksi dengan label sebenarnya, sementara *precision* menunjukkan ketepatan model dalam mendeteksi ancaman. *Recall* mencerminkan kemampuan model dalam menangkap semua ancaman yang ada, dan *F1-score* merupakan kombinasi harmonis antara *precision* dan *recall* yang memberikan gambaran seimbang terhadap kinerja model secara keseluruhan. Persamaan accuracy, *precision*, *recall*, dan *F1-score* dapat dilihat pada Rumus 1-4 [24].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (3)$$

$$Precision = \frac{TP}{TP+FP} \qquad (4)$$

$$Recal = \frac{TP}{TP+FN} \qquad (5)$$

$$F1-score = 2 . \frac{Precision.Recall}{Precision+Recall} \qquad (6)$$

Keterangaan:

- *True Positive* (TP) : model berhasil memprediksi serangan ketika data tersebut memang attack.
- *True Negative* (TN) : Model memprediksi data sebagai normal atau suspicious dan kenyataannya memang bukan serangan.
- *False Positive* (FP) : model salah memprediksi data sebagai *attack*, padahal data tersebut sebenarnya *normal* atau *suspicious*.
- *False Negative* (FN): model memprediksi data sebagai *normal* atau *suspicious*, padahal data tersebut sebenarnya adalah *attack.*

3. HASIL DAN PEMBAHASAN

Hasil penelitian ini menggambarkan proses deteksi dan klasifikasi serangan siber dengan metode machine learning algoritma *K-Nearest Neighbor* (KNN) dan *Random Forest* (RF) berdasarkan data log sistem yang diperoleh dari Biro Sistem Informasi Universitas Muhammadiyah Purwokerto.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan dataset log serangan siber yang diperoleh dari Biro Sistem Informasi Universitas Muhammadiyah Purwokerto. Dataset ini memuat berbagai informasi penting terkait aktivitas lalu lintas jaringan yang mencurigakan dan potensi ancaman siber yang terdeteksi oleh sistem keamanan internal. Dataset tersebut dapat dilihat pada Gambar 2.

```
# Load data
import pandas as pd #Import pandas in the current notebook or section
data = pd.read_csv('SERANGAN50010.csv', sep=';')

# Tampilkan informasi data
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            500 non-null    object
 1   Type            500 non-null    object
 2   Protocol        500 non-null    object
 3   Method          500 non-null    object
 4   URL/Directory   500 non-null    object
 5   Src Zone        500 non-null    object
 6   Src IP          500 non-null    object
 7   Src Location    500 non-null    object
 8   Src Port        500 non-null    int64
 9   Xff_IP          500 non-null    object
 10  Dst Zone        500 non-null    object
 11  Dst IP          500 non-null    object
 12  Dst Port        500 non-null    int64
 13  Rule ID         500 non-null    object
 14  State Code      500 non-null    object
 15  Policy Name     500 non-null    object
 16  Description     500 non-null    object
 17  Rule Name       500 non-null    object
 18  Solution        500 non-null    object
 19  Reference       500 non-null    object
 20  SANGFOR WIKI    405 non-null    object
 21  Threat Level    500 non-null    object
 22  Action          500 non-null    object
 23  Impact          500 non-null    object
 24  Data Packet     500 non-null    object
dtypes: int64(2), object(23)
memory usage: 97.8+ KB
None
```

Gambar 2. Dataset Penelitian

Dataset yang digunakan terdiri atas 500 baris dan 25 atribut, yang sebagian besar berisi data lengkap. Namun, kolom *URL/Directory* dan SANGFOR WIKI mengandung data hilang dan perlu diproses lebih lanjut. Sebagian besar atribut bertipe kategorikal, sedangkan *Src Port* dan *Dst Port* bertipe numerik. Informasi dalam dataset ini mencakup aspek lalu lintas dan keamanan jaringan yang penting untuk deteksi serta klasifikasi serangan siber. Penjelasan dari setiap atribut dapat dilihat pada Tabel 1.

Tabel 1. Penjelasan Tiap Atribut

No	Nama Atribut	Pengertian Atribut
1	*Date*	Waktu dan tanggal kejadian atau serangan tercatat.
2	*Type*	Jenis serangan atau aktivitas yang terdeteksi.
3	*Protocol*	Protokol jaringan yang digunakan (HTTP, HTTPS, dll).
4	*Method*	Metode HTTP yang digunakan (GET, POST, dll).
5	*URL/Directory*	URL atau direktori target dari permintaan.
6	*Src Zone*	Zona sumber lalu lintas (biasanya TRUST atau UNTRUST).
7	*Src IP*	Alamat IP sumber (penyerang).
8	*Src Location*	Lokasi geografis dari alamat IP sumber.
9	*Src Port*	Port yang digunakan oleh sumber.
10	*Xff_IP*	Alamat IP yang diteruskan oleh header *X-Forwarded-For*.
11	*Dst Zone*	Zona tujuan lalu lintas.
12	*Dst IP*	Alamat IP tujuan (target serangan).

p-ISSN: 2775-4227
e-ISSN: 2775-4219

No	Nama Atribut	Pengertian Atribut
13	*Dst Port*	Port yang digunakan oleh tujuan.
14	*Rule ID*	ID aturan keamanan yang mendeteksi ancaman.
15	*State Code*	Kode status dari sistem keamanan.
16	*Policy Name*	Nama kebijakan atau aturan yang diterapkan.
17	*Description*	Deskripsi umum tentang serangan yang terdeteksi.
18	*Rule Name*	Nama aturan atau rule spesifik dalam sistem keamanan.
19	*Solution*	Solusi atau rekomendasi penanganan terhadap ancaman.
20	*Reference*	Referensi tambahan atau catatan terkait ancaman.
21	*Sangfor Wiki*	Tautan ke dokumentasi Sangfor Wiki terkait serangan.
22	*Threat Level*	Tingkat ancaman (Rendah, Sedang, Tinggi).
23	*Action*	Tindakan yang diambil oleh sistem (misalnya: *Deny*).
24	*Impact*	Dampak atau konsekuensi dari serangan.
25	*Data Packet*	Isi data paket yang terkait dengan permintaan atau serangan.

3.2 Pra-Pemrosesan

```
# 6. Encode SEMUA FITUR dan TARGET pakai LabelEncoder (aman dan praktis)
from sklearn.preprocessing import LabelEncoder

# Ambil fitur yang kamu sebutkan
x = df[['Type', 'Protocol', 'Method', 'Src Zone', 'Src Location']]

# Inisialisasi encoder
encoder = LabelEncoder()

# Encode semua fitur
X_encoded = x.apply(lambda col: encoder.fit_transform(col.astype(str)))

# Target (label) hanya satu: Threat Level
y = df['Threat Level']
y_encoded = encoder.fit_transform(y.astype(str))
```

```
# 7. Cek distribusi kelas sebelum balancing
unique, counts = np.unique(y_encoded, return_counts=True)
plt.bar(unique, counts, color='orange')
plt.xlabel('Kelas (encoded)')
plt.ylabel('Jumlah data')
plt.title('Distribusi Kelas pada Target Sebelum Oversampling')
plt.show()
```

Gambar 3. Dataset Penelitian

Gambar 3 menunjukkan tahapan awal dalam proses praprosesan data yang dimulai dengan melakukan seleksi atribut dari total 25 atribut yang tersedia dalam dataset. Dari seluruh atribut tersebut, dipilih lima atribut yang dianggap paling relevan untuk proses klasifikasi, yaitu *Type, Protocol, Method*, dan *Src Location* sebagai fitur (variabel independen), serta *Threat Level* sebagai target klasifikasi (variabel dependen). Pemilihan atribut ini dilakukan untuk menyederhanakan model serta memfokuskan analisis pada informasi yang paling berkontribusi terhadap penentuan level ancaman dalam data. Setelah atribut-atribut tersebut ditentukan, dilakukan proses konversi data kategorikal ke bentuk numerik menggunakan LabelEncoder dari pustaka *scikit-learn*, baik untuk keempat fitur maupun targetnya, guna memastikan data dapat dibaca oleh algoritma pembelajaran mesin yang digunakan.

Proses *encoding* penting untuk menjaga konsistensi format data dan menghindari kesalahan dalam pemodelan. Setelah proses *encoding* selesai, langkah berikutnya adalah melakukan visualisasi distribusi kelas pada target *Threat Level* dalam bentuk grafik batang. Visualisasi ini bertujuan untuk mengetahui sebaran data pada masing-masing kelas sebelum dilakukan proses *balancing* data, karena ketidakseimbangan distribusi kelas dapat menyebabkan bias dalam pembelajaran model dan menurunkan performa klasifikasi.

3.3 Pemodelan

Penelitian ini menerapkan dua algoritma klasifikasi yang populer dalam pembelajaran mesin, yaitu *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF), guna mengevaluasi kemampuannya dalam memprediksi dan mengklasifikasikan data serangan siber berdasarkan log sistem yang tersedia.

1. *K-Nearest Neighbors* (KNN)

Model *K-Nearest Neighbors* (KNN) pada Gambar 4 dibangun menggunakan KNeighborsClassifier dengan *n_neighbors* = 5, yang berarti model mempertimbangkan lima tetangga terdekat dalam menentukan kelas suatu data. Sebelum pelatihan, data dinormalisasi menggunakan *StandardScaler* karena KNN sensitif terhadap skala

p-ISSN: 2775-4227
e-ISSN: 2775-4219

data. Proses pelatihan dilakukan dengan data latih yang telah diskalakan (*X_train* dan *y_train*), lalu digunakan untuk memprediksi label pada data uji (*X_test*) yang juga telah dinormalisasi. Pemilihan jumlah tetangga bertujuan untuk mencapai keseimbangan antara kompleksitas model dan akurasi prediksi.

```
# 11. KNN
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train_scaled, y_train)
y_pred_knn = knn.predict(X_test_scaled)
```

Gambar 4. KNN Model

2. *Random Forest* (RF)

Model *Random Forest* dibangun menggunakan RandomForestClassifier dengan 100 pohon keputusan (*n_estimators=100*) dan *random_state=42* untuk memastikan hasil yang konsisten. Tidak seperti KNN, algoritma ini tidak memerlukan proses *scaling*, sehingga langsung dilatih menggunakan data latih (*X_train* dan *y_train*) dan digunakan untuk memprediksi data uji (*X_test*). Model *Random Forest* ditunjukkan pada Gambar 5.

```
# 12. Random Forest (tidak perlu scaling)
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
```

Gambar 5. *Random Forest* Model

Berdasarkan hasil pemodelan, metode K-Nearest Neighbor (KNN) dan *Random Forest* memiliki pendekatan yang berbeda dalam melakukan prediksi. KNN mengandalkan kedekatan jarak antar data untuk menentukan kelas, sehingga performanya sangat dipengaruhi oleh distribusi data dan pemilihan parameter k. Sementara itu, *Random Forest* bekerja dengan menggabungkan beberapa pohon keputusan yang dibangun dari subset data yang berbeda untuk menghasilkan prediksi yang lebih akurat.

3.4 Evaluasi

Evaluasi performa dilakukan terhadap dua model klasifikasi yang digunakan, yaitu *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF). Evaluasi pada Gambar 6. dilakukan dengan menggunakan metrik akurasi dan *classification report* yang mencakup *precision*, *recall*, dan *f1-score*. Kode Evaluasi *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF) dapat dilihat pada Gambar 6.

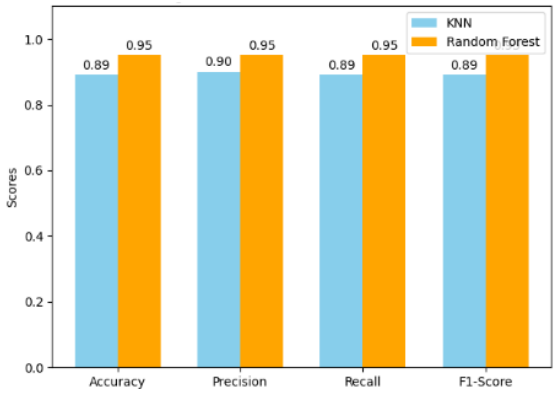
```
# 13. Evaluasi
print("=== KNN Evaluation ===")
print("Accuracy:", accuracy_score(y_test, y_pred_knn))
print(classification_report(y_test, y_pred_knn))

print("\n=== Random Forest Evaluation ===")
print("Accuracy:", accuracy_score(y_test, y_pred_rf))
print(classification_report(y_test, y_pred_rf))
```

```
=== KNN Evaluation ===
Accuracy: 0.8931623931623932
              precision    recall  f1-score   support

           0       0.91      0.86      0.88       112
           1       0.88      0.93      0.90       122

    accuracy                           0.89       234
   macro avg       0.90      0.89      0.89       234
weighted avg       0.89      0.89      0.89       234


=== Random Forest Evaluation ===
Accuracy: 0.9487179487179487
              precision    recall  f1-score   support

           0       0.91      0.99      0.95       112
           1       0.99      0.91      0.95       122

    accuracy                           0.95       234
   macro avg       0.95      0.95      0.95       234
weighted avg       0.95      0.95      0.95       234
```

Gambar 6. Evaluasi *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF)

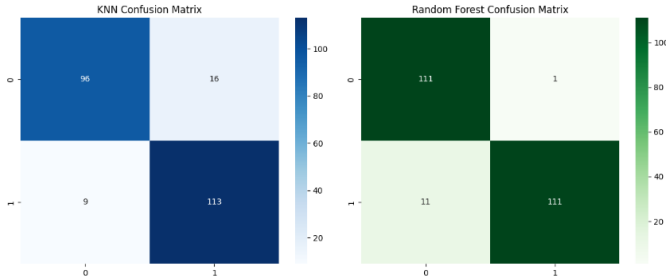
p-ISSN: 2775-4227
e-ISSN: 2775-4219

Hasil grafik performa serta metrik evaluasi dari model K-Nearest Neighbors (KNN) dan Random Forest (RF) dapat dilihat pada Gambar 7 dan Gambar 8. Pada Gambar 7 ditampilkan grafik perbandingan akurasi dan metrik lainnya untuk kedua model, sedangkan Gambar 8 menyajikan confusion matrix yang menggambarkan detail prediksi masing-masing model. Kedua gambar ini memberikan gambaran komprehensif mengenai efektivitas dan keakuratan model dalam klasifikasi data.



Gambar 7. Grafik peforma *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF)

Berdasarkan hasil evaluasi, akurasi model *Random Forest* mencapai 94,87%, lebih tinggi dibanding KNN yang hanya memperoleh 89,32%. Perbedaan akurasi ini dapat disebabkan oleh kemampuan *Random Forest* dalam melakukan *ensemble learning*, yaitu menggabungkan banyak pohon keputusan untuk menghasilkan prediksi yang lebih stabil dan akurat, serta lebih tahan terhadap *overfitting*.



Gambar 8. Metrik evaluasi *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF)

Sementara itu, KNN cenderung sensitif terhadap data yang memiliki distribusi tidak merata atau *noise*, karena klasifikasi dilakukan berdasarkan kedekatan jarak antar titik data di ruang fitur tanpa proses pelatihan model yang mendalam. Selain akurasi, metrik evaluasi lainnya juga menunjukkan keunggulan *sss*. Model ini mencatat *f1-score* sebesar 0.95 pada kedua kelas, sedangkan KNN hanya mencapai 0.88 untuk kelas 0 dan 0.90 untuk kelas 1. Meskipun KNN cukup seimbang dalam *precision* dan *recall*, *Random Forest* tetap lebih unggul dalam ketepatan klasifikasi.

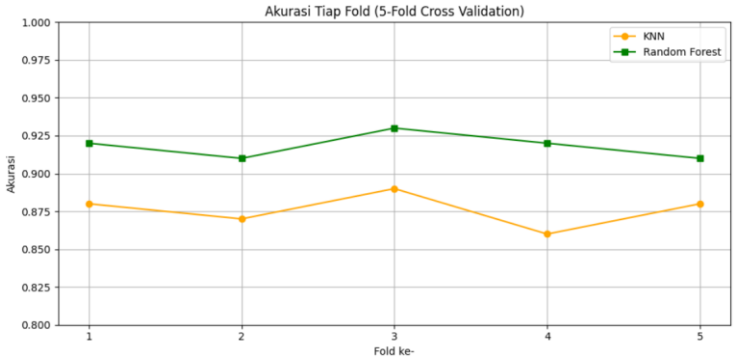
Untuk meningkatkan reliabilitas evaluasi model dan mengurangi kemungkinan bias akibat pembagian data tunggal (single train-test split), dilakukan validasi silang menggunakan metode K-Fold Cross Validation. Teknik ini membagi data menjadi lima subset (fold) yang secara bergantian digunakan sebagai data uji dan latih, sehingga memberikan estimasi performa model yang lebih stabil dan representatif. Proses ini diterapkan pada kedua algoritma, yaitu K-Nearest Neighbors (KNN) dan Random Forest (RF), dengan menggunakan data yang telah melalui tahap encoding, oversampling dengan SMOTE, serta normalisasi. Hasil dari validasi silang ini menunjukkan bahwa Random Forest tetap menunjukkan performa unggul dengan akurasi rata-rata sebesar 95,53%, sementara KNN memperoleh akurasi sebesar 93,24%, seperti ditampilkan pada Gambar 9.

```
=== K-Fold Cross Validation ===
Akurasi KNN: 0.9324
Akurasi Random Forest: 0.9553
```

Gambar 9. Hasil K-Fold Cross Validation terhadap Akurasi KNN dan Random Forest

p-ISSN: 2775-4227
e-ISSN: 2775-4219

Gambar 10 menampilkan perbandingan akurasi masing-masing algoritma pada setiap fold selama proses 5-Fold Cross Validation. Terlihat bahwa Random Forest secara konsisten menghasilkan akurasi yang lebih tinggi dan stabil di tiap fold dibandingkan KNN. Hal ini semakin memperkuat temuan sebelumnya bahwa Random Forest lebih unggul dalam hal konsistensi performa pada data yang divalidasi silang.



Gambar 9. Grafik Akurasi Tiap Fold pada 5-Fold Cross Validation untuk KNN dan Random Forest

4. KESIMPULAN

Pesatnya perkembangan serangan siber menuntut sistem deteksi yang cerdas dan adaptif untuk mengamankan jaringan informasi. Untuk menjawab tantangan tersebut, dilakukan evaluasi terhadap dua algoritma klasifikasi, yaitu *K-Nearest Neighbors* (KNN) dan *Random Forest* (RF), guna mengetahui metode yang paling efektif dalam mendeteksi ancaman melalui data log jaringan. Hasil pengujian menunjukkan bahwa *Random Forest* memiliki akurasi lebih tinggi sebesar 94,87% dibandingkan KNN yang hanya mencapai 89,32%, serta nilai *precision, recall,* dan *f1-score* yang lebih stabil pada masing-masing kelas. Meski demikian, KNN memiliki keunggulan dalam kesederhanaan dan interpretabilitas, yang masih relevan untuk sistem dengan sumber daya terbatas. RF dinilai lebih unggul dalam efektivitas klasifikasi, namun bersifat lebih kompleks dan kurang transparan dibanding KNN. Ke depan, pengujian dengan teknik validasi silang dan eksplorasi model lain, termasuk pendekatan *deep learning*, dapat menjadi langkah lanjutan untuk meningkatkan keandalan dan adaptivitas sistem deteksi serangan siber.

DAFTAR PUSTAKA

[1] J. Lee, J. Kim, I. Kim, and K. Han, "Cyber Threat Detection Based on Artificial Neural Networks Using Event Profiles," *IEEE Access*, vol. 7, pp. 165607–165626, 2019, doi: 10.1109/ACCESS.2019.2953095.

[2] A. B. Setiawan, "Peningkatan Keamanan Supervisory Control and Data Acquisition (Scada) Pada Smart Grid Sebagai Infrastruktur Kritis," *J. Penelit. Pos dan Inform.*, vol. 6, no. 1, p. 59, 2016, doi: 10.17933/jppi.2016.060104.

[3] M. K. Hussein, N. Bin Zainal, and A. N. Jaber, "Data security analysis for DDoS defense of cloud based networks," *2015 IEEE Student Conf. Res. Dev. SCOReD 2015*, pp. 305–310, 2015, doi: 10.1109/SCORED.2015.7449345.

[4] N. Anwar and I. Riadi, "Analisis Investigasi Forensik WhatsApp Messanger Smartphone Terhadap WhatsApp Berbasis Web," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 3, no. 1, p. 1, 2017, doi: 10.26555/jiteki.v3i1.6643.

[5] M. H. Rifai, D. A. Pramudya, and R. R. Narfandi, "Analisis peran teknologi kecerdasan buatan dalam mengoptimalkan proses deteksi terhadap serangan siber," pp. 495–502, 2024.

[6] Z. Liu, X. Li, and D. Mu, "Intelligent Analysis and Prediction of Computer Network Security Logs Based on Deep Learning," *Electron.*, vol. 13, no. 22, pp. 1–15, 2024, doi: 10.3390/electronics13224556.

[7] M. P. Aji, "Klasifikasi Tingkat Ancaman Siber menggunakan Pembelajaran Mesin pada Web Application Firewall (WAF) Cyber," *J. Media Pratama*, vol. 17, no. 1, pp. 61–73, 2023.

[8] S. N. Adzimi, H. A. Alfasih, F. N. G. Ramadhan, S. N. Neyman, and A. Setiawan, "Implementasi Konfigurasi Firewall dan Sistem Deteksi Intrusi menggunakan Debian," *J. Internet Softw. Eng.*, vol. 1, no. 4, p. 12, 2024, doi: 10.47134/pjise.v1i4.2681.

[9] D. Ferarizki, Yusra, M. Fikry, F. Yanto, and F. Insani, "Klasifikasi Sentimen Masyarakat di Twitter Terhadap Ancaman Resesi Ekonomi 2023 dengan Metode K-Nearest Neighbor," *J. Ekon. Vol. 18, Nomor 1 Maret201*, vol. 2, no. 1, pp. 41–49, 2020.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

[10] I. Maulana and Alamsyah, "Optimalisasi Deteksi Serangan DDoS Menggunakan Algoritma Random Forest, SVM, KNN dan MLP pada Jaringan Komputer," *Indones. J. Math. Nat. Sci.*, vol. 45, no. 1, pp. 1–8, 2022.

[11] Sunaryono, "Penelitian Komparasi Algoritma Klasifikasi," vol. 1, no. 1, pp. 1–12, 2017.

[12] J. C. W. Chan and D. Paelinckx, "Evaluation of *Random Forest* and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery," *Remote Sens. Environ.*, vol. 112, no. 6, pp. 2999–3011, 2008, doi: 10.1016/j.rse.2008.02.011.

[13] W. Ghozi, F. A. Rafrastara, R. R. Sani, and U. D. Nuswantoro, "Deteksi Serangan Denial of Service (DoS) dan Spoofing pada Internet of Vehicles menggunakan Algoritma K-Nearest Neighbor (KNN)," vol. 6, no. 2, 2024.

[14] S. Kurniawan, W. Gata, D. A. Puspitawati, N. -, M. Tabrani, and K. Novel, "Perbandingan Metode Klasifikasi Analisis Sentimen Tokoh Politik Pada Komentar Media Berita Online," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 176–183, 2019, doi: 10.29207/resti.v3i2.935.

[15] N. M. Lutimath, C. Chethan, and B. S. Pol, "Prediction of heart disease using machine learning," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 10, pp. 474–477, 2019, doi: 10.35940/ijrte.B1081.0982S1019.

[16] S. Adi and A. Wintarti, "Komparasi Metode Support Vector Machine (Svm), *K-Nearest Neighbors* (Knn), Dan *Random Forest* (Rf) Untuk Prediksi Penyakit Gagal Jantung," *MATHunesa J. Ilm. Mat.*, vol. 10, no. 2, pp. 258–268, 2022, doi: 10.26740/mathunesa.v10n2.p258-268.

[17] M. Putri, "Penerapan K-Optimal pada Algoritma Modified K–Nearest Neighbor (Mk-Nn) untuk Klasifikasi Kelulusan Mahasiswa (Studi Kasus : Teknik Informatika Uin Suska Riau)," *J. Ekon. Vol. 18, Nomor 1 Maret201*, vol. 2, no. 1, pp. 41–49, 2020.

[18] F. M. N. Akbar, "Metode KNN (K-Nearest Neighbor) untuk Menentukan Kualitas Air," *J. Tekno Kompak*, vol. 18, no. 1, p. 28, 2024, doi: 10.33365/jtk.v18i1.3241.

[19] F. Akbar, S. Achmadi, and A. Mahmudi, "Implementasi Analisis Data Kredit Nasabah Menggunakan Metode K-Nearest Neighbors," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 4, no. 1, pp. 82–92, 2020, doi: 10.36040/jati.v4i1.2351.

[20] Y. Religia, A. Nugroho, and W. Hadikristanto, "Klasifikasi Analisis Perbandingan Algoritma Optimasi pada *Random Forest* untuk Klasifikasi Data Bank Marketing," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, pp. 187–192, 2021, doi: 10.29207/resti.v5i1.2813.

[21] A. N. Pratiwi and E. Utami, "Prediksi Kinerja Akademik Matematika Siswa berdasarkan Kepribadian Big Five menggunakan *Random Forest* dengan Teknik Synthetic Minority Over - Sampling Personality Traits using *Random Forest* with Synthetic Minority Over - Sampling," vol. 14, pp. 985–1000, 2025.

[22] S. Rahayu, J. J. Purnama, A. B. Pohan, F. S. Nugraha, and S. Nurdiani, "Prediction Of Survival Of Heart Failure Patients Using Random Forest," *J. PILAR Nusa Mandiri*, vol. 16, no. 2, pp. 255–260, 2020.

[23] Leny Margaretha Huizen and Roy Rudolf Huizen, "Optimalisasi Keamanan IoT dan Edge Computing Menggunakan Model Machine Learning," *J. Sist. dan Inform.*, vol. 17, no. 2, pp. 89–94, 2024, doi: 10.30864/jsi.v17i2.543.

[24] M. A. A. R. Asif *et al.*, "Performance evaluation and comparative analysis of different machine learning algorithms in predicting cardiovascular disease," *Eng. Lett.*, vol. 29, no. 2, pp. 731–741, 2021.